UNIVERSITY OF MARYLAND
COMPUTER SCIENCE CENTER

COLLEGE PARK, MARYLAND

49
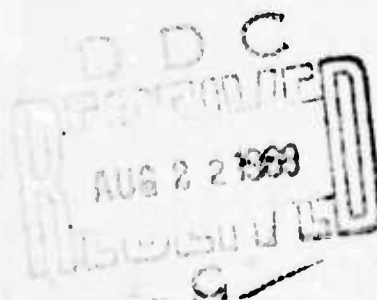
Technical Report 69-96   August, 1969
Nonr-5144(00)

Separable graphs, planar graphs
and web grammars

by

G. Ugo Montanari

RESEARCH PROGRESS REPORT

<u>Title</u>: "Separable graphs, planar graphs and web grammars",
G. U. Montanari, University of Maryland Computer
Science Center Technical Report 69-96, August 1969;
Contract Nonr-5144(00).

<u>Background</u>: The Computer Science Center of the University
of Maryland is investigating the theory of image
processing by computer. One area under study is the
theory of "grammars" whose "languages" are sets of
labelled directed graphs.

<u>Condensed Report Contents</u>: This paper is concerned with
the class of "web grammars", introduced by Pfaltz
and Rosenfeld, whose languages are sets of labelled
graphs. A slightly modified definition of web gram-
mar is given, in which the rewriting rules can have
an applicability condition, and it is proved that in
general, this extension does not increase the gener-
ative power of the grammar. This extension is use-
ful, however, for otherwise it is not possible to
incorporate negative contextual conditions into the
rules, since the context of given vertex can be un-
bounded. A number of web grammars are presented
which define interesting classes of graphs, includ-
ing unseparable graphs, unseparable planar graphs
and planar graphs. All the grammars in this paper
use "normal embeddings" in which the connections be-
tween the web that is written and the host web are
conserved, so that any rewriting rule affects the
web only locally.

<u>For Further Information</u>: The complete report is available
in the major Navy technical libraries and can be
obtained from the Defense Documentation Center. A
few copies are available for distribution by the
author.

1. Introduction

A major problem in picture interpretation and description is the extension of the basic ideas and methods developed for string grammars to structures which are interconnected in more general ways. A number of different approaches have been studied in the literature (see the review paper [1]). The most recent, and also the most general and flexible theory has been introduced by Pfaltz and Rosenfeld in [2]. They apply the rewriting procedure typical of string grammar to general labelled graphs or "webs". Examples of web grammars are given in [2] for trees, series-parallel networks, Pascal triangles and other interesting classes of graphs.

The extension of string language theory to webs is by no means trivial; there are special problems associated with it. Examples are the "embedding" of the rewritten webs [2], and the impossibility of incorporating negative contextual conditions into the rules. This latter fact is due to the unboundness of the context of a vertex in a graph (see Section 2 of this paper).

The web grammar approach has several potential advantages. Graph theory constitutes a well developed and well studied field, in which, perhaps with some generalization, the various notations introduced in the picture grammar literature can be unified. At the same time, the possibility of deriving families of graphs by rewriting initial graphs according to finite sets of rules may be useful to graph theorists as well,

because this method both standardizes and generalizes the induction procedures frequently used in graph theory. Important properties of the derived class of graphs can often be proved by simple inspection of the rules, as in the case of the Euler relation for plane graphs (see Section 4).

Section 2 of this paper discusses the definition of a web grammar. In particular, it is shown that the addition of an "applicability condition" to the rules of the grammar does not change the generative power of the grammar, because an equivalent grammar without applicability conditions can always be found. A new class of web grammars, called "monotone web grammars", that seems to be more general than context-sensitive web grammars, is then introduced. (The proof of equivalence of the monotone and context-sensitive classes of grammars, given by Chomsky in [3] in the case of string languages, does not apply to web languages.) The concept of indirect generation of a language is then introduced. Both these generalizations allow more freedom in designing web grammars, while still leaving it decidable whether or not a given web belongs to the language of a given web grammar.

In Section 3 a grammar for nonseparable graphs is first presented. The notion of separability web of a graph is then introduced, and a grammar for generating such webs is shown. Finally, a grammar for generating the graphs having a given separability tree is given. Section 4 gives grammars for various classes of planar graphs; some of them simultaneously generate a dual graph. Finally, it is shown that the four color conjecture for planar graphs can be reduced to the equivalence of the languages of two web grammars.

2. Web Grammars

In this section, we introduce web grammars and the related terminology. Our definition is very similar to that of Pfaltz and Rosenfeld; the only difference is that we allow a rewriting rule to be applied only if a specified condition is satisfied. However, the introduction of applicability conditions is only a matter of convenience in expressing contextual restrictions. In fact, we will prove that the generative power of web grammars is not increased in this way.

Let V be a finite set, which we will call the vocabulary; the elements of V will be called symbols. A directed web W over V is a triple (N_W, F_W, A_W) where N_W is a set of vertices, F_W is a labelling function from N_W into V, and A_W is a set of ordered pairs of elements of N_W which are called arcs. If V has only one element, webs are clearly equivalent to graphs. Given a web $W = (N_W, F_W, A_W)$ over V, the web $S = (N_S, F_S, A_S)$ over the same V is called a subweb of W if:

a) N_S is a subset of N_W
b) $F_S(X) = F_W(X)$ if $X \in N_S$
c) A_S consists of just those pairs in A_W whose terms are both in N_S.

Undirected webs are defined analogously; they can be regarded as a special case of directed webs in which, between any pair of vertices P and Q, either no arc exists, or both of the arcs (P, Q) and (Q, P) are present. All of the web grammars defined in the later sections of this paper are for undirected webs.

A _web grammar_ G is a finite entity which allows us, in general, to define an infinite set of webs, called the language of G. Formally, G is a triple (V, I, R). V is the vocabulary; it consists of two disjoint parts, a _nonterminal_ vocabulary V_N and a _terminal_ vocabulary V_T. I is a finite set of _initial webs_ over V. R is a finite set of _rewriting rules_. Every rewriting rule, if applicable, specifies what changes must be made to a given web. Formally, a rewriting rule is a quadruple (α, C, β, E) where α, β are webs, C is a logical function called the _contextual condition_ of the rule, and E is a set of logical functions called the _embedding_ of the rule. The rule is applicable to a web W if α is a subweb of W and C is true. The effect of the application of the rule is to replace a subweb α of W with the web β. The logical functions of E specify the embedding of β in $W-\alpha$; that is, they specify whether or not each vertex of $W-\alpha$ is connected to each vertex of β. Note that E must be well-defined for every web W to which the rule is applicable. Both C and E will typically depend on the labels, or the number, of vertices of W connected with some vertex of α; however, in general they can depend on the entire W and on the particular subweb α of W which is to be rewritten.

The _language_ L_G generated by G consists of those webs on V_T that can be derived from the initial webs by applying the rewriting rules any number of times, in any order. Two grammars are _equivalent_ if they generate the same language. For example, the following web grammar G(V, I, R), with one symbol and one re-

writing rule, has as language the set of directed binary
trees with least point:

$$V_N = \emptyset; \quad V_T = \{t\}; \quad V = \{t\}$$
$$I = \{\overset{\bullet}{t}\}$$

$R = \left\{\left(\begin{matrix} t \\ \overset{\bullet}{L}_1 \end{matrix}\right.\right.$; in W there must not exist more than one

arc from L_1; $\quad \underset{R_1 \quad R_2}{t \longrightarrow t}$; all the vertices of W that were
connected with L_1 are connected with R_1, and no vertex
is connected with $R_2\left.\left.\right)\right\}$. In fact, it can be immediately
proved by induction that in any web generated by this
grammar, there is a unique path from the initial point
to any other point. Furthermore, the applicability
condition prevents the creation of vertices from which
there are arcs to more than two vertices. Conversely,
any binary tree with k+1 vertices can be derived from
a suitable tree with k vertices by adding one arc and
one vertex.

The embeddings used in all the rules considered in
this paper will always be of the same type as that in
the example just given: an arc between α and W-α is
never created or deleted, but simply shifted to a vertex
of β. How the shifting is done will be specified with
the aid of indices assigned to the vertices of α and β;
we have denoted these by letters of the form L_i (ith
vertex of the left member of the rule) and R_i. The
embedding is defined by a function from the set of
vertices of α into the set of vertices of β. If a
vertex P of β is the image of a collection of vertices
of α, all of the arcs between W-α and these vertices

of α in the original web W become arcs to or from P in the rewritten web. If a vertex of β is not the image of any vertex of α, there are no arcs between it and W-α in the rewritten web. A useful special case is that in which the function is one-to-one (as it is in the binary tree example); such an embedding will be called **normal**. If all the rules of a web grammar G have normal embedding, the grammar will be called normal.

As remarked at the beginning of this section, we have introduced the function C in order to allow an explicit statement of contextual conditions (*) on the application of a rewriting rule. If the desired condition is a "positive" one (for example: a vertex of W-α with a given label must be connected by an arc to a given vertex of α), it is very easy to embody this context in the rule by enlarging its left member (see [2], pp. 614-616). On the other hand, if the contextual condition is "negative", as in our binary tree example, there is no way of expressing it by enlarging the webs of the rule. This is one of the important differences between string grammars and web grammars. In fact, in string languages, the number of symbols which can precede or follow a given symbol is finite, since the the alphabet is finite. Thus a rule forbidding a symbol can be replaced by a finite set of rules requiring each of the other symbols. On the other hand, in web languages, there is no limit on the number of

---

(*)   If one uses a web grammar for **parsing** webs rather than generating them, the contextual conditions become embeddings (and vice versa); this is another reason why contextual conditions are desirable.

vertices which can be connected with a given vertex,
so that a finite number of rules cannot in general be
sufficient. In simple cases a slightly more compli-
cated grammar without negative contextual conditions
can often be constructed. In the general case, the
following theorem shows how contextual conditions can
always be included in the embedding part of the rule,
so that the equivalence to the Pfaltz and Rosenfeld
definition is proved. However, the resulting grammars
are somewhat tricky, and the normal embedding, if
present, is lost.

Theorem 1. Given any web grammar G, a web grammar H
equivalent to G can always be found, such that the
rules of H have no applicability conditions (i.e., the
logical functions C are always true).

Proof. Let $G = (V_G, I_G, R_G)$ and $V_G = V_{NG} \cup V_{TG}$. The
terminal vocabulary of H must obviously be equal to
V_{TG}. The nonterminal vocabulary of H consists of V_{NG}
together with an extra symbol $*$. The initial webs of
H are obtained from the webs of I_G by adding two
isolated vertices labelled $*$ to each of them. The
rules of H are as follows:

a) A set of rules that can be obtained from the rules
   of G by the following procedure: An isolated
   vertex labelled $*$ is added to the left and right
   members. Logical function C is always true. The
   embedding is the same, as far as vertices not
   labelled $*$ are concerned. Furthermore, an arc is
   added from the vertex labelled $*$ of the right

member to the other vertex labelled * if the con-
textual condition of the corresponding rule of G
is not satisfied, or if the two vertices labelled *
were already connected.

b) The rule * * → ∅

If web $W \in L_G$, then $W \in L_H$. In fact, given any deri-
vation of W in G, it is possible to derive W by also
applying first the corresponding rules of H and finally
rule (b). Conversely, if $W \in L_H$ then $W \in L_G$. In fact,
given any derivation of W in H, rule (b) must have been
applied exactly once, because no vertex labelled with
the nonterminal symbol * is created in any rule, and
furthermore it must be the last rule applied, because
all the other rules require a *-vertex. Moreover, any
time a rule of H has been applied, the condition for
the application of the corresponding rule of G must
have been satisfied, since otherwise an arc between the
two vertices labelled * would have been created, which
would never have been removed and which would have pre-
vented the application of rule (b). Thus a parallel
derivation of W in G can be found. //

Different classes of web grammars can be defined
according to what restrictions are imposed on the re-
writing rules. If for every rule (rewriting α into β)
a vertex P exists in α such that $\alpha -\{P\}$ is a proper
subweb of β (i.e., one point only is rewritten, and
not erased), and if the connections between $\alpha -\{P\}$ and
$W-\alpha$ are not changed by E, the web grammar is called
context-sensitive. If we compare context sensitive
web grammars with context sensitive string grammars

we see that the latter are interesting because they
are very general, and at the same time they have the
following remarkable properties ([3], pp. 143-144):

a) the derivation of any string of the language is
    representable by a directed tree;

b) the problem of whether or not a given string be-
    longs to a given language is solvable.

The second property is particularly important
for practical purposes. It follows directly from
the fact that under application of the rules, the
lengths of strings are monotonically non-decreasing,
so that only a finite number of derivations need be
tested for any given finite length string. For
string grammars, Chomsky ([3], p. 145) has proved
also the converse: if all the rules of a grammar
G are of the type $\varphi \rightarrow \Psi$ with Ψ at least as long as
φ, then it is possible to construct a context sensi-
tive grammar G' equivalent to G. In the grammar G'
a set of rules is substituted for every rule of G, such
that if the first rule of the set is used, all the
others must be applied in sequence, or else a termi-
nal string cannot be achieved. The same type of
reasoning does not seem to work for web grammars.
In fact, if a rule of G applies to two different sub-
webs α' and α'' of a web W, the sequences of rules of
G' applied to α' and α'' may "interfere". Thus con-
text-sensitive web grammars clearly have the essential
properties (a) and (b) above, but they seem to be
comparatively less powerful. In some grammars that

we report in this paper, we have found it convenient to
use only the restriction that in any rule, β has no fewer
vertices than α. Such grammars will be called <u>monotone</u>.(*)
As mentioned above, this condition is sufficient for
assuring decidability. In some cases this restriction
too is strong, because it is desirable to create
"auxiliary" vertices in the derivation (for example, in
Theorem 1), and in a monotone grammar they cannot be
erased. On the other hand, in some cases the "auxiliary"
vertices have a special significance cf their own; for
instance, in the grammar for planar graphs given in
Section 4, they constitute a "dual graph". In these
cases we prefer to preserve the monotonicity of the
grammar, and to consider the desired webs as subwebs
of the webs generated by the grammar. More formally,
we say that a web language L is <u>indirectly generated</u>
by a grammar G if:

a) the vocabulary V_L of L is a subset of the terminal
   vocabulary V_{TG} of G;

b) the language L conists of just the <u>subwebs</u> of the
   terminal webs generated by G whose vertices are
   labelled with symbols belonging to V_L:

c) in any web generated by G, in which N vertices are
   labelled with symbols of V_L, the number of vertices
   labelled with symbols of $V_G - V_L$ cannot exceed a fixed
   value M_N.

Note that property (c) assures that it is always decidable
whether a given web belongs to a language indirectly
generated by a given grammar. For instance, in Thecrem

(*) Note that a normal grammar is always monotone.

1, if G is context sensitive (monotone), a context
sensitive (monotone) grammar without applicability
conditions that indirectly generates L_G can be found
by simply substituting for rule (b) of the grammar H
the following rule:

$$\overset{*}{\cdot} \quad \overset{*}{\cdot} \to \underset{\cdot}{t} \quad \underset{\cdot}{t}$$

where t is the only terminal symbol of H that does
not belong to V_L.

A special case of context sensitive web grammars
is that of context free web grammars. In such a gram-
mar, there are no applicability conditions on the rules,
and the left member of each rule consists of only one
vertex.

3. Separable and nonseparable graphs

A connected graph (*) G is called separable [4] if it has two subgraphs F_1 and F_2, having at least one arc, such that every arc of G belongs to either F_1 or F_2, while F_1 and F_2 have exactly one vertex V in common. Vertex V will be called a cut vertex. If two such subgraphs cannot be found, the G is called nonseparable. Nonseparable graphs can be also defined by means of a web grammar, as proved in the following theorem:

Theorem 2: The context sensitive, normal web grammar of Fig. 1, with one terminal symbol and without applicability conditions, generates exactly all connected nonseparable webs (**).

Proof: Only nonseparable webs. Whitney ([5], Th. 6) has proved that a necessary and sufficient condition for a graph to be separable is that three distinct vertices A,B,C exist such that every (simple) path from B to C passes through A. Reasoning by induction, both initial webs are nonseparable. Furthermore, if a web is nonseparable before the application of any rule, it is also nonseparable afterwards, because the application of any rule does not erase any possible path.

All nonseparable webs. Conversely, Whitney ([5], Th. 19) has shown that it is possible to build up any connected nonseparable graph containing more than two arcs by

---

(*) In this and in the next section we will always be concerned with connected graphs having at least one arc, and without loops.

(**) The above definitions relative to graphs can be immediately extended to webs.

starting with a circuit and adding to it arcs or chains
of arcs. In our grammar, the only nonseparable graph
with one arc belongs to the initial set, while an
initial circuit of any length can be constructed from
the initial triangle by recursive application of rule
2. The addition of an arc is performed by rule 1,
and the addition of a chain of any length can be ob-
tained by applying first rule 1 and then rule 2,
recursively. //

If a graph G is separable, its components F_1 and
F_2 can be either separable or not; if they are separable,
they can in turn be broken up into components, and so
on. Since the sum of the numbers of arcs of the com-
ponents is equal, at any stage, to the number of arcs
of the entire graph, we must eventually reach a situ-
ation in which all the components are nonseparable.
These subgraphs of G are called the nonseparable com-
ponents of G. Whitney ([5], Th. 12) has proved that
the nonseparable components do not depend on the order
in which the decomposition takes place. This decom-
position can be represented as a partition of the arcs
of the graph; a component is a subgraph that contains
exactly all the arcs of some element of the partition.
Cut vertices belong to the vertex sets of at least two
components. The components are "connected" at cut
vertices, but they never form a "circuit of graphs"(*)
([5], Th. 17). A graph will be called arccomposite

(*) That is, it is not possible to find a cyclic se-
quence of components such that any one of them has
just one cut vertex in common with its predecessor
and its successor.

if at least one of its components is a one-arc graph,
i.e., consists of two vertices and one arc connecting
them; nonarccomposite, otherwise. Note that a non-
separable graph is arccomposite only if it is a one-arc
graph.

Given any connected graph, we define the separa-
bility web of the graph as follows: The vocabulary
is {i, c}. Vertices labelled "i" are internal vertices
while vertices labelled "c" are connection vertices.
Internal vertices are in correspondence with the non-
separable components of the graph, while connection
vertices correspond to cut vertices. An arc is traced
between a connection vertex and an internal vertex of
the separability web if the corresponding cut vertex
belongs to the corresponding nonseparable component.
The separability web of a given graph is unique, as
an immediate consequence of the uniqueness of the
decomposition.

The following theorem characterizes, and gives
a web grammar for, separability webs:
Theorem 3: a) A web on the vocabulary {i, c} can be
regarded as a separability web if and only if no
circuit of arcs is present, every arc connects vertices
labelled with different symbols, and every vertex which
is connected with only one other vertex is labelled
"i". b) The normal, context free grammar in Fig. 2
generates just the set of all separability webs.
Proof: a) Only if part. This part is a direct con-
sequence of the definition.

If part. Let N be the maximum number of vertices con-
nected with any i-vertex.[*] A complete graph on N
vertices is associated with every i-vertex, and a
one-to-one correspondence is defined between some
vertices of this graph and all the c-vertices of the
web connected with the i-vertex. A graph G is then
constructed by coalescing the vertices of the complete
graphs   correspond to the same c-vertices of the
web. The complete subgraphs are nonseparable, and
by hypothesis no circuit of graphs is present. There-
fore, according to Whitney ([5], Th. 17(1), (3)), the
complete subgraphs are the nonseparable components
of G. Thus the given web is the separability web of
a graph G.

b) Webs generated by the grammar in Fig. 2 clearly
satisfy the hypothesis of part (a) and therefore they
are separability webs. Conversely, any separability
web can be generated in this way. In fact, reasoning
by induction, assume that the grammar generates all
the separability webs with k i-vertices. Given a
separability web W with k+1 i-vertices, an i-vertex
P connected to only one c-vertex must exist - for,
no c-vertex is of degree 1, and in a graph without
circuits at least one vertex of degree 1 must
exist. Let Q be the c-vertex connected to P. If Q
is of degree greater than 2, then W - {P} is a
separability web with k i-vertices and can be gen-
erated by the grammar. The last rule applied to Q
in the derivation of W - {P} must be rule 4. Applying

(*) If this number is 1, then let N = 2.

the rules 3 and 2 just before this rule, it is possible
to derive W as well. If Q is of degree 2, let R be
the other i-vertex connected to Q. The web W - {P, Q}
is a separability web with k i-vertices and can be
generated by the grammar. The last rule applied to
R must be rule 2. Applying the rules 1, 4 and 2 just
before this rule it is possible to derive W as well. //

We now give a more complicated example of a web
grammar.

Theorem 4: Given any separability web (having more
than one vertex) as initial web, the monotone, normal
web grammar G in Fig. 3 indirectly generates all the
separable graphs having the given initial web as
separability web. [Note that rule 10 in this grammar
has a "negative" applicability condition. The language
directly generated by G consists of the separable graphs
with two extra vertices, labelled b and a, for every
nonseparable component. The b-vertices are connected
with all the vertices of the components, while a-vertices
are connected only with the corresponding b-vertices.]

Proof: **All separable graphs, etc.** Given an I-vertex
P of the initial web, if the corresponding component
is a one-arc graph, and if P is connected to one c-
vertex (two c-vertices), rule 2 (rule 4) must be
applied. If the component corresponding to P has more
than one arc and one, or two, or more than two c-vertices,
rules 1, 3, or 5, respectively, must be used first.
From the initial triangle, any nonseparable component
with more than one arc can then be derived; the method
is the same as that proved for the grammar in Fig. 1.

Rule 8 in Fig. 3 is the equivalent of rule 1 in Fig. 1,
while rule 6 or 7 is the equivalent of rule 2, depend-
ing on whether the new vertex that must be inserted into
the component is a cut vertex or not. Note that the
connections of the c-vertices with the B-vertex assure
that rules 6, 7 and 8 are applied to vertices belonging
to the same component. Furthermore, in rule 6, the
connection with the inserted vertex is shifted from the
A-vertex to the B-vertex of the component. When the
entire component has been built up, as proved in Theorem
2, all the corresponding cut vertices have been inserted,
and thus rules 9 and 10 can be applied, because no vertex
is now connected to the A-vertex of the component.
Only separable graphs, etc. Let us consider a derivation
of a web W. To any I-vertex of the initial web, one of
the rules 1-5 must have been applied. If rule 2 or 4
was used, rules 6, 7 and 8 cannot have been applied to
the b-vertex, so that the final rule 10 must have con-
cluded the generation of a one-arc component. If rule
1, 3 or 5 was used, any application of rules 6, 7 or 8
preserves the nonseparability of the component, as proved
by Theorem 2. When finally rules 9 and 10 apply to con-
vert the nonterminal symbols A and B into the terminal
symbols a and b, rule 6 must have already inserted all
the cut vertices of the component, since otherwise the
applicability condition of rule 10 would not be satisfied. //

It is very easy to see that if we do not allow in-
directness, no monotone grammar exists for this language.
In fact, there exist graphs which have fewer vertices than

their separability webs. For instance, the separability web of the graph ●━━●━━●━━● is $\underset{\text{I}\quad\text{c}\quad\text{I}\quad\text{c}\quad\text{I}}{●━━●━━●━━●━━●}$.

4. Planar graphs

A graph is called <u>planar</u> if it can be mapped onto
the plane (or on the sphere: [5] p. 355). The regions
into which it divides the plane will be called <u>meshes</u>.
This definition is topological in character; however,
some purely combinatorial equivalent definitions can
be found. Kuratowsky [6] has proved that a graph is
planar if and only if it contains no subgraphs having
either of two specific forms. Whitney [5] has proved
that a graph is planar if and only if it has a "dual".
MacLane [4] has shown that a graph is planar if and
only if a collection of "basic" circuits can be found,
such that every arc of the graph is considered in this
collection exactly twice. In this section, we give a
characterization of planar graphs as indirectly gen-
erated by a monotone web grammar (or, as remarked in
Section 2, as directly generated by a web grammar
which allows vertex deletion).

Whitney's definition of dual is clearly independent
of the particular mapping of the graph on the plane.
However, his definition of dual coincides ([5], Th. 30)
with the well-known definition in which a vertex of
the dual corresponds to a mesh of the given plane (*) con-
nected graph, and two vertices of the dual are connected
with an arc if the two corresponding meshes are adjacent
to the same arc. This construction provides a one-to-one
correspondence between the arcs of the plane graph and

(*) A planar graph is called <u>plane</u> if it is considered
as mapped on the plane.

those of the dual graph, and between the vertices of the plane graph and the meshes of the dual. Furthermore, the dual is plane and the dual of the dual is the given plane graph. Some difficulties arise in this definition of dual if only graphs without multiple edges are considered. This is the case if, as in our definition of a web in Section 2, arcs are defined as pairs of vertices. This definition is the most natural, and the most often used in graph theory. However, if some vertex of the graph is of degree two, i.e., if some arcs are serially connected, the above construction gives a dual with multiple edges. Thus, unless otherwise stated, we consider only plane graphs and webs without serially connected arcs. However, our results could be easily extended if a definition allowing multiple edges were given. On the other hand, no problem exists if the same mesh is adjacent to one arc on both sides; a loop is then generated in the dual. This happens in particular if a vertex of degree one exists in the graph.

Given a plane graph, for every vertex P it is possible to specify a cyclic sequence of meshes by examining, in counterclockwise order, the meshes adjacent to P. Two successive meshes of the sequence are both adjacent to an arc connected to P. Note that this sequence can contain the same mesh more than once, possibly even in successive positions.

We can now prove the following theorem:

Theorem 5: a) A connected plane graph G is separable if and only if a vertex P exists, such that its sequence

of meshes contains the same mesh more than once. Fur-
thermore, P is a cut vertex of G.

b) A plane graph G is arccomposite if and only if an
arc (P, Q) exists which is adjacent to the same region
on both sides.

Proof: a) If the same mesh is present more than once
in the sequence of meshes of vertex P, it means that
a closed curve C passing through P can be traced, which
does not intersect any other vertex or arc of the graph,
and which partitions the graph in two subgraphs with at
least one arc, having only the vertex in common. Thus,
by definition, G is separable and P is a cut vertex.
Conversely, if G is separable, it can be constructed by
letting two vertices of two disconnected graphs coalesce
in P, so that a curve C as above can be found.

b) The one-arc subgraph S is a nonseparable component
of G if and only if no circuit exists in G to which
the arc of S belongs ([4], p. 24). This is equivalent
to saying that a closed curve intersecting only (P, Q)
can be found. //

Given a plane graph, assume that a simple closed
plane curve exists with the following properties:

a) The curve intersects the plane graph only at vertices.
[As a consequence of this first property, the segment of
curve between two adjacent intersection vertices belongs
entirely to a single mesh, and the number of intersection
vertices and meshes is equal.]

b) Different segments of the curve belong to different meshes.

If such a curve exists, it will be called a
cut curve of the plane graph. Note that property (b)
is dual to the simplicity of the curve.

Given a plane graph and a cut curve, the vertices
and meshes of the graph are evidently partitioned into
three classes: <u>internal</u>, <u>boundary</u> and <u>external</u> vertices
and meshes, while only internal and external arcs can
be distinguished. We can now prove the following Lemma.

<u>Lemma 1</u>: Given a nonarccomposite plane graph G with-
out loops, and a cut curve, it is always possible to
find an internal arc with the following properties:

a) One vertex and one mesh adjacent to the arc are
    a boundary vertex and a boundary mesh (*);

b) the other vertex (the graph is without loops) is
    either an internal vertex or a boundary vertex
    adjacent to the first vertex;

c) the other mesh (the graph is nonarccomposite) is
    either an internal mesh, or a boundary mesh adjacent
    to the first mesh.

<u>Proof</u>: At least one arc satisfying condition (a) is
connected with every boundary vertex, since otherwise
two adjacent segments of the cut curve would belong
to the same mesh. Furthermore, at least one of these
arcs (and actually two, if more than one internal arc
exists at that vertex) is adjacent to a boundary mesh,
as is easy to see by examining the sequence of meshes
of the vertex. Conditions (b) and (c) too are clearly
satisfied by some of these arcs if the number of bound-
ary vertices (and meshes) is three or less, because in
this case the first vertex (mesh) is adjacent to all
the other vertices (meshes). Now we assume that the
number of boundary vertices is four or more, and that

(*) Clearly the mesh belongs to the sequence of meshes
     of the vertex.

no arc satisfying condition (a) also satisfies conditions
(b) and (c). If an arc satisfies (a) but does not sat-
isfy (b), it means that it connects two nonadjacent
boundary vertices. Let (P, Q) be an arc which satisfies
condition (a) (i.e., P is a boundary vertex and the first
mesh M is a boundary mesh) but which does not satisfy
condition (c) because the second mesh N is a boundary
mesh not adjacent to M. Let R and S be the boundary
vertices that are the endpoints of the segment of the
cut curve belonging to N. Thus R ≠ P and S ≠ P, and
either R or S (or both), say R, is not adjacent to P,
since if P, R, S were pairwise adjacent they would have
to be the only three boundary vertices. Therefore mesh
N is adjacent to P and to a vertex R not adjacent to P.
Thus a new plane graph can be obtained by adding the
arc (P, R) to G. In conclusion, if condition (b) or
(c) is not satisified, a new plane graph G' can be con-
structed in which every boundary vertex is connected by
an internal arc with at least one nonadjacent boundary
vertex. (*)

We will now show that G' cannot be plane, so that
we have reached a contradiction. Fig. 4 shows this
fact in the case of four boundary vertices. On the
other hand, it is possible to prove that if such a
plane graph G' exists in the case of n (n > 4) bound-
ary vertices, it exists also for n-1 vertices. In fact,
given a plane graph G' with n boundary vertices, if
redundant arcs are erased, any arc (P, Q) is the only

(*) If it is condition (b) which fails to hold for every
     boundary vertex, no construction is necessary.

internal arc connected with either P or Q (or both),
say P. If R and S are the boundary vertices adjacent
to P, vertex Q cannot be adjacent to both of them (other-
wise n = 4). Let Q be not adjacent to R. Therefore
the vertex P can be erased and the arc (P, Q) can be
replaced by the arc (R, Q), obtaining a plane graph
with (n - 1) boundary vertices. //

We can now give web grammars for various classes
of planar graphs. The simplest one is introduced by
the following theorem.

Theorem 6: The normal, monotone web grammar of Fig. 5
indirectly generates exactly all nonarccomposite planar
graphs. More precisely, this grammar directly generates
exactly all nonarccomposite planar graphs and their
duals (without serially connected arcs, or arcs in
parallel; see the remarks at the beginning of this
section).(*)

Proof: **All planar graphs, etc.** Given any such graph,
let us consider a plane representation of this graph.
We shall construct a copy of this graph and of its
dual by applying the rules of the grammar in Fig. 5.
We will prove by induction that it is possible to find
a sequence of cut curves such that the external part of
the graph represents the part already copied, and that
it is possible to construct the next step in the copy
by applying a rule of the grammar. More precisely, at
a given stage of application of the grammar, the ver-
tices of the copy that correspond to external vertices
are labelled "a", the boundary vertices are labelled
"A", the vertices of the dual which correspond to ex-

(*) If a planar graph has more than one plane representation,
and thus more than one dual (i.e., if it is not triply
connected), all the possible duals can be derived with
this grammar.

ternal meshes are labelled "b", and boundary mesh dual
vertices are labelled "B". All external arcs and the
corresponding arcs in the dual are present in the copy
web. The cut curve is represented in the copy web by
a circuit of arcs connecting the A- and B-vertices.
Every A-vertex (B-vertex) is connected neither with
any A-vertex (B-vertex) nor with any b-vertex (a-ver-
tex), but it is connected with exactly two B-vertices
(A-vertices) and possibly with some a-vertices (b-ver-
tices). The first cut curve is found as crossing the
two adjacent vertices and the two adjacent meshes of
any arc adjacent to the unbounded mesh of the plane
graph. Furthermore this first cut curve must contain
all the graph, except the initial arc. Correspondingly,
rule 1 is applied to the initial web. Now, given any
cut curve, assume that the above described web has
already been constructed. Then, according to Lemma 1,
an internal arc can be found, of one of the following
four types:

a) one vertex and one mesh adjacent to the arc are on
    the boundary, the other vertex and the other mesh
    are internal.

b) the two meshes of the arc are adjacent boundary
    meshes, one vertex is a boundary vertex, the other
    is an internal vertex.

c) two adjacent boundary vertices, one boundary mesh
    and one internal mesh

d) two adjacent boundary vertices and two adjacent
    boundary meshes.

In the above cases (a, b, c, d), grammar rules 2, 3, 4,
5 respectively apply. Rule 6 must be applied instead
of rule 5 if only one internal arc was present. It
can be immediately verified that the new A-B circuit
corresponds to the cut curve obtained by adding the
arc under consideration to the internal arcs, while
the copy still satisfies the above characterization.
In Fig. 6 we see the A-B circuit and the cut curve
superimposed, before and after the application of the
rules. Note that the new cut curve never passes twice
through the same vertex or mesh, because the new ver-
tices and meshes just introduced cannot, by the choice
of the new arc, have been boundary vertices and meshes
of the old cut curve. Every application of a rule of
the grammar introduces an arc into the copy. Thus
for a graph with n arcs it is necessary to apply the
rules of the grammar exactly n times. At the end,
i.e., after application of rule 6, two disconnected
webs have been constructed: a copy of the given graph,
labelled with "a", and a copy of its dual, labelled
with "b".

Only planar graphs, etc. We will prove by induction that
during the application of the grammar, the graph and its
dual can be mapped onto the plane. Let us assume, in
fact, that at a given stage, a web W has been generated
in which the vertices labelled A and B form a circuit,
with A's and B's alternating. All the other vertices,
labelled a and b, if any, are outside. Let us consider
now a web W' that is obtained by connecting all the
A-vertices of W with an internal vertex, labelled c,

and connecting with an arc the pairs of B-vertices adjacent to the same A-vertex in the circuit. (If there are no A's and B's, the c-vertex need not be added, so that W' is the same as W.) We assume that the subgraph on the vertices of W' labelled c, A or a is a nonarccomposite plane graph G, while the subgraph on the vertices labelled B or b corresponds to its dual. Clearly, the circuit of A- and B-vertices is a cut curve of G. These assumptions are certainly true if only rule 1 has been applied to the initial web. Now apply any of rules 2-6 to the web W. It is immediate to verify that the above characterizations of webs W and W' is applicable to the new web. In particular, rule 2 increments by one each the number n of A- and B-vertices, rules 3 and 4 do not change n, rule 5 decrements n by one, while rule 6 decrements it by two. However, any rule allows the new arc it introduces to be mapped into the plane. Thus the new graph G is still plane, and is still nonarccomposite, because the added arc is adjacent to two different meshes (Theorem 5(b)). But at the last step, there can be no A's and B's left, so that W' is the same as W, and G is the graph which was indirectly generated by the grammar. //

Note that every rule has nonterminal symbols in its right member, except rule 6. Thus rule 6 must be the last rule used. But if rule 6 applies, the A-B circuit is eliminated, so that it can be applied only once.

Fig. 7 shows an example of the application of the grammar in Fig. 5 to the simplest possible nonarccomposite

planar graph. Here the rules applied, in succession,
to obtain stages (a-f) were (1), (2), (4), (3), (5),
(6). Other derivations of this graph are also possible.

Using this grammar it is very easy to prove the
Euler relation between vertices, meshes and arcs for
the generated planar graphs. In fact, let n_B, n_a and
n_b be the numbers of the vertices labelled B, a and b
in a web W generated by the grammar at some step, and
let us define the function $f = n_B + n_a + n_b$. It is
easy to verify by simple inspection that f increases
by one if rules 2-5 are applied, and increases by two
if rules 1 or 6 are applied. But rules 1 and 6 are
applied exactly once, and the application of any rule
introduces one arc. Thus, if e is the total number
of arcs, for every terminal web we have $e + 2 = n_a + n_b$,
i.e., the Euler relation.

By modifying the grammar of Fig. 5, it is easy to
obtain the grammar for nonseparable planar graphs shown
in Fig. 8. This grammar is obtained from the grammar in
Fig. 5 by introducing a new terminal symbol c. Then in
rule 2, instead of simply disconnecting two boundary ver-
tices, they are disconnected but at the same time are both
connected to a c-vertex. The applicability condition of
rule 5 assures that boundary vertices which have already
been connected and then disconnected will not be connected
again. The proof that this grammar actually works is
given in the following Lemma and Theorem.

Lemma 2: Let us consider any derivation of a nonarc-
composite planar graph and of its dual according to the
grammar of Theorem 6. The vertices of the graph (of the

dual) are labelled A(B) when they are on the boundary,
and are rewritten with the terminal symbol a(b) when
they become internal vertices. Given an A-vertex P of
the graph, a B-vertex Q of the dual has been connected
to and disconnected from P during the derivation ex-
actly as many times as the mesh corresponding to Q ap-
pears in the sequence of meshes of P.

Proof: In the application of any rule of the grammar
of Theorem 6 an A-vertex and a B-vertex appear together
in the right or left web of the rule if and only if the
arc introduced by this rule is connected to the A-vertex
and is adjacent to the mesh corresponding to the B-vertex.
In fact, in Theorem 6 it has been shown that in the de-
rivation of any web the two arcs introduced by any rule
are corresponding arcs of a plane graph and of its dual.
On the other hand, in a plane nonarccomposite graph, if
a mesh M appears n times in the mesh sequence of a ver-
tex P, then there exist exactly 2n arcs connected to P
and adjacent to M. In fact, M cannot be adjacent to
the same arc on both sides, by Theorem 5(b). Thus in
any derivation of this graph, the vertex P and the ver-
tex Q of the dual corresponding to M have appeared in
the same rewritten web exactly 2n times. Note now that
if an A- and a B-vertex are disconnected (or any of
them does not exist) before the application of any rule,
then they are connected afterwards, and if they are
connected before the application of any rule, then they
are disconnected (and possibly relabelled) afterwards.
Therefore P and Q have been connected and disconnected
exactly n times. //

Theorem 7: The normal, monotone grammar of Fig. 8 indirectly generates all nonseparable planar graphs.

Proof: According to Theorem 5(a-b), the class of nonseparable planar graphs is obtained from the class of nonarccomposite planar graphs by eliminating all the graphs with the same mesh adjacent twice to the same vertex, and adding the one-arc graph. In the grammar of Fig. 8, this graph is added as an initial graph, while the applicability condition of rule 5, according to Lemma 2, does not allow the derivation of graphs with the same mesh adjacent twice to the same vertex. //

By adding some simple rules to the grammar of Theorem 6, it is possible to obtain a grammar that generates all planar graphs, as proved in the following theorem.

Theorem 8: The normal, monotone grammar of Fig. 9 indirectly generates exactly all planar graphs.

Proof: **All planar graphs.** Given a planar graph G, for each cut vertex, let us consider the union of all the components having more than one arc and having the given cut vertex in common; this is a subgraph of G. We obtain in this way the decomposition of G into nonarccomposite pieces and one-arc graphs. Now let us assume that there is at least one nonarccomposite piece N in this decomposition; otherwise G is a tree and can be obtained from the initial web $\overset{a}{\bullet}\underline{\quad}\overset{a}{\bullet}$ by applying rule 7, as in the grammar for binary trees in Section 2. N can be constructed using the initial web $\overset{S}{\bullet}$ and rules 1 - 6.

All the one-arc graphs (*) having a cut vertex in common
with N can then be constructed using rule 7 or 8.
Rule 8 must be used (case (a)) if the other vertex of
the one-arc graph is a cut vertex belonging to a non-
arccomposite piece, rule 7 if the other vertex does not
belong to such a piece, but is either (case (b)) a cut
vertex common to one or more other one-arc graphs or
(case (c)) is not a cut vertex. In case (a), a new non-
arccomposite piece can be derived using rules 1 - 6,
and in case (b) the new one-arc components can be de-
rived using rule 7 This procedure can be iterated
until all the components have been derived. In fact,
the graph is connected, and if some component would
have more than one cut vertex in common with the al-
ready derived components, a circuit of graphs would be
present, which would constitute a single nonseparable
component ([5], Th. 16).

Only planar graphs. The application of rule
7 to any terminal or nonterminal web surely leaves
planar the subweb consisting of all the a-vertices.
The application of rule 8 generates a new S-vertex,
from which a new nonarccomposite component can be
generated. The derivation of this component does not
interfere with the derivation of any other component,
because the A-B circuit generated for this component
has no vertex in common with the A-B circuit of any other compo-

---

* Note that, by construction, only one-arc components can
  have a common vertex with N.

nent. This property can be proved by induction: it is
true for the first A-B circuit generated by rule 1,
and rules 2 - 6 apply only to A- and B-vertices be-
longing to the same circuit. Thus the assertion fol-
lows from the fact that if the components of a graph
are planar, the graph is planar ([5], Th. 27). //

An interesting class of problems in graph theory
is concerned with the least number of colors necessary
for coloring the vertices of the graphs of a given family
in such a way that every arc connects vertices of dif-
ferent color. An especially challenging problem arises
in the case of planar graphs. For this family of graphs,
five colors have been proved sufficient, and four colors
necessary; the so-called four color conjecture says that
four colors are also sufficient. The proof of this con-
jecture can easily be shown equivalent to determining if
the languages of two web grammars G_1 and G_2 are equal.
G_1 is the grammar in Fig. 5 (*). In G_1, every rule is
concerned with a pair of A-vertices which can possibly
be generated or relabelled by the rule. In any case, the
arc created by the rule connects either A-vertices or cor-
responding relabelled vertices. G_2 is derived from G_1
by the following procedure. Instead of the nonterminal
symbol A, four symbols A_1, A_2, A_3 and A_4 are used..
Twelve rules are substituted for each of the rules 2,
3 and 5, such that all the possible pairs of different
symbols among the four symbols A_1 A_2 A_3 and A_4 replace

(*) It is obvious that if any nonarccomposite planar
graph were four colorable, the same would be true
for any planar graph.

the pair of A symbols. Only six rules need to be sub-
stituted for each of the rules 1, 4, 6, because of
symmetry. The language of this second grammar is
clearly the class of nonarccomposite four-colorable
planar graphs (*). In fact, any derivation in G_2 has
a parallel derivation in G_1 and defines a coloring of
the terminal graph, while given a derivation in G_1, and
a coloring of the terminal graph, a derivation in G_2
can be found.

(*) In general, by using n new symbols, the class of non-
arccomposite n-colorable planar graphs is obtained.

References

[1] Miller, F. W. and Shaw, A. C., Linguistic Methods in Picture Processing - A Survey, Proc. Fall Joint Computer Conf., December 1968, 279-290.

[2] Pfaltz, J. L. and Rosenfeld, A., Web Grammars, Proc. International Joint Conference on Artificial Intelligence, May 7-9, 1969, Washington, D. C., 609-619.

[3] Chomsky, N., On Certain Formal Properties of Grammars, Information and Control, 2 (1), 1959, 139-167.

[4] MacLane, S., A Combinatorial Condition for Planar Graphs, Fund. Math., 28, 1937, 22-32.

[5] Whitney, H., Non-separable and Planar Graphs, Trans. Amer. Math. Soc., 34, 1932, 339-362.

[6] Kuratowsky, C., Sur le problème des courbes gauches en Topologie, Fund. Math., 15, 1930, 271-283.

Figure captions

Fig. 1 - This normal, context sensitive web grammar
         generates exactly all the nonseparable graphs

Fig. 2 - This normal, context-free web grammar generates
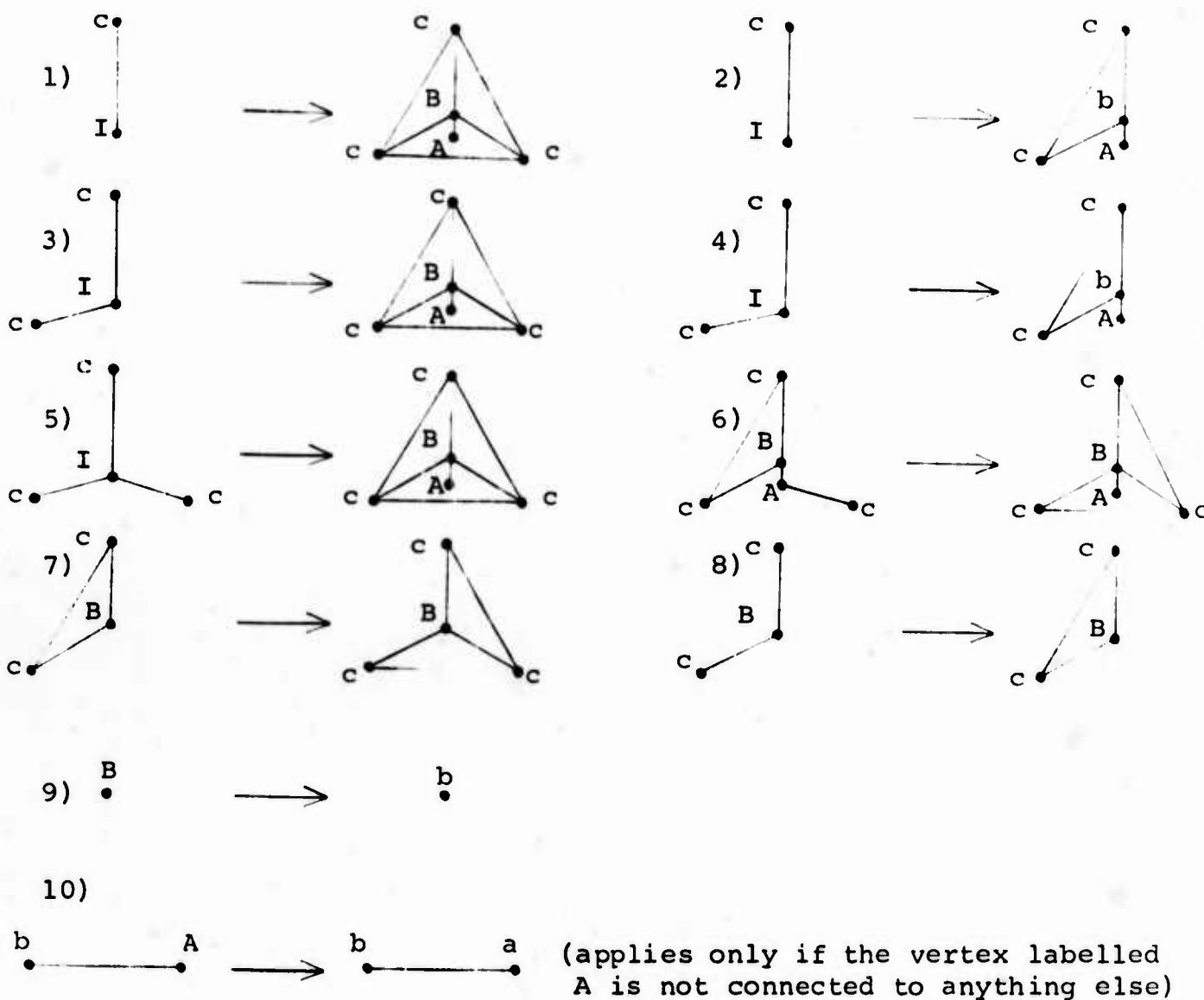   exactly all the separability webs

Fig. 3 - This normal, monotone web grammar, given a
         separability web as initial web, indirectly
         generates exactly all the separable graphs
         having the given web as separability web.
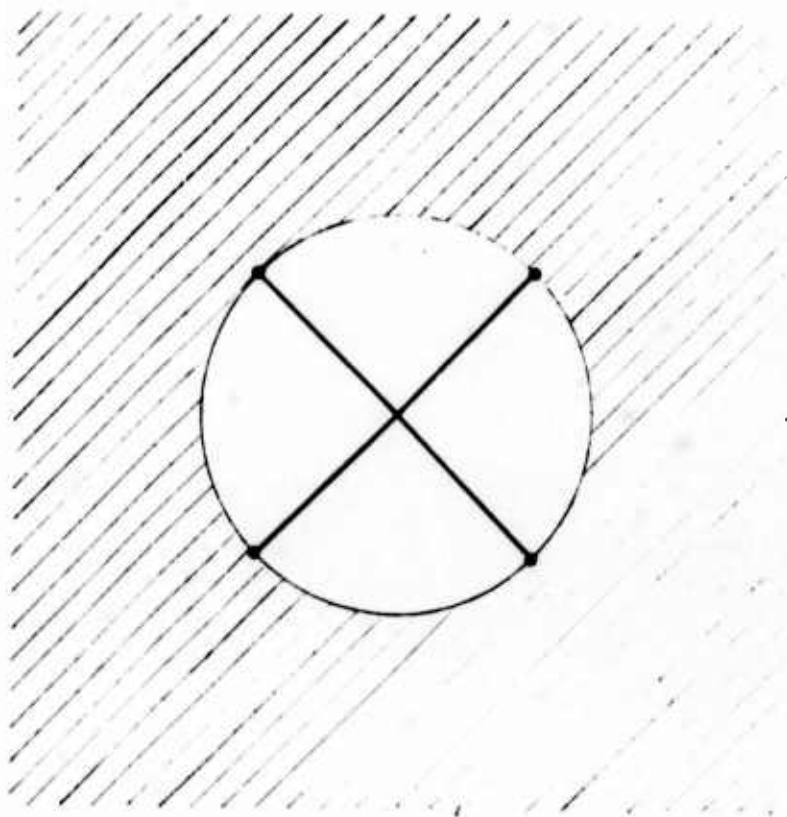   Note the applicability condition on rule 10

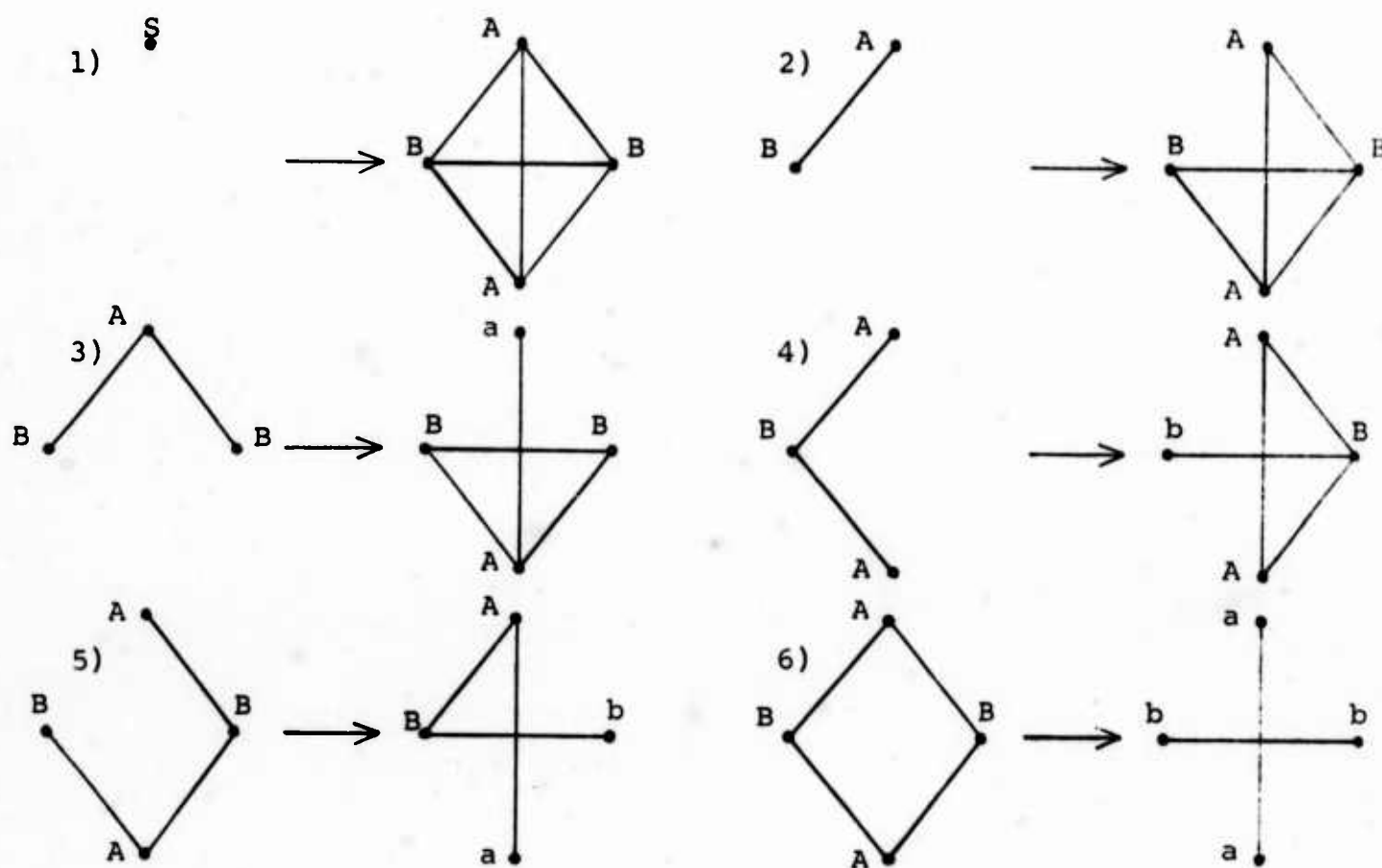Fig. 4 - This graph cannot be planar

Fig. 5 - This normal, monotone web grammar indirectly
         generates exactly all nonarccomposite planar
         graphs

Fig. 6 - Superposition of the A-B circuit and of the
         cut curve in applying the rules of the grammar
         in Fig. 5

Fig. 7 - An example of application of the web grammar
         in Fig. 5
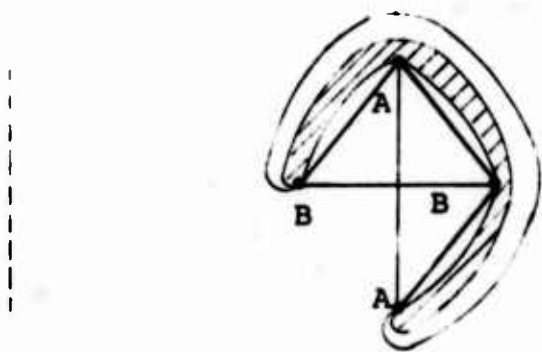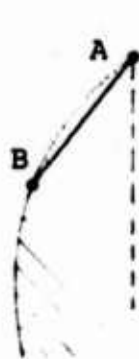
Fig. 8 - This normal, monotone web grammar indirectly
         generates exactly all nonseparable planar
         graphs. Note the applicability condition on
         rule 5

Fig. 9 - This normal, monotone web grammar indirectly
         generates exactly all planar graphs

$V_N = \{A\}$; $V_T = \{t\}$

$I = \left\{ \begin{array}{c} \end{array} \right.$ $, \underset{\bullet\quad\bullet}{t\quad t} \left. \right\}$

Rewriting rules:

1)

2)

3)

Fig. 1. This normal, context sensitive web grammar
generates exactly all the nonseparable graphs

Note: In all of the figures, the vertices that correspond
under the normal embedding are shown in correspond-
ing positions.

$V_N = \{S, A\}$; $V_T = \{i, c\}$

$I = \{\underset{\bullet}{c}\}$

Rewriting rules:

1)

2)

3)

4)

Fig. 2. This normal, context free web grammar generates exactly all the separability webs

$$V_N = \{I, A, B\} \quad ; \quad V_T = \{a, b, c\} \quad ; \quad V_L = \{c\}$$

$I = \{$any separability web; internal vertices labelled I, cut vertices labelled c$\}$

Rewriting rules:

Fig. 3. This normal, monotone web grammar, given a
separability web as initial web, indirectly
generates exactly all the separable graphs
having the given web as separability web.
Note the applicability condition on rule 10.

Fig. 4.   This graph cannot be planar

$V_N = \{S, A, B\}$; $V_T = \{a, b\}$; $V_L = \{a\}$

$I = \{S\}$

Rewriting rules:

Fig. 5. This normal, monotone web grammar indirectly
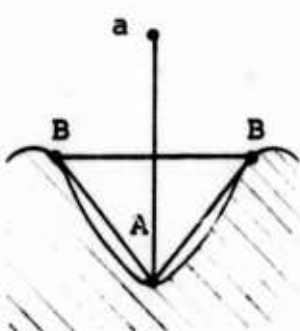generates exactly all nonarccomposite planar
graphs
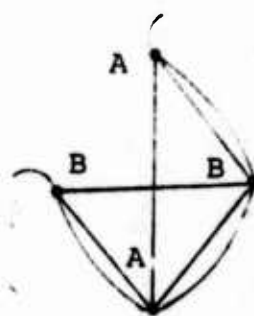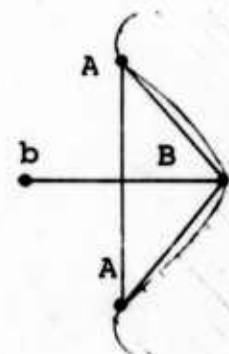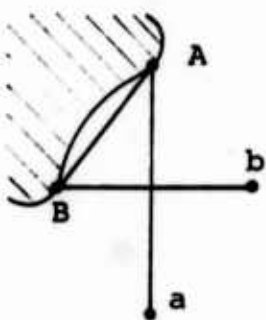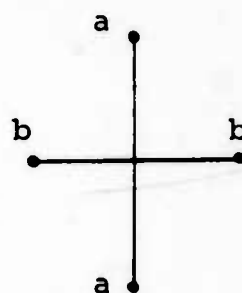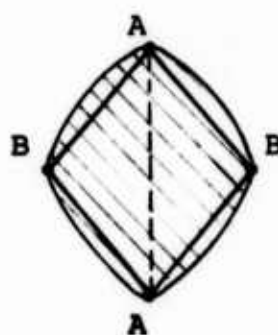
Fig. 6. Superposition of the A-B curcuit and
of the cut curve in applying the rules
of the grammar in Fig. 5
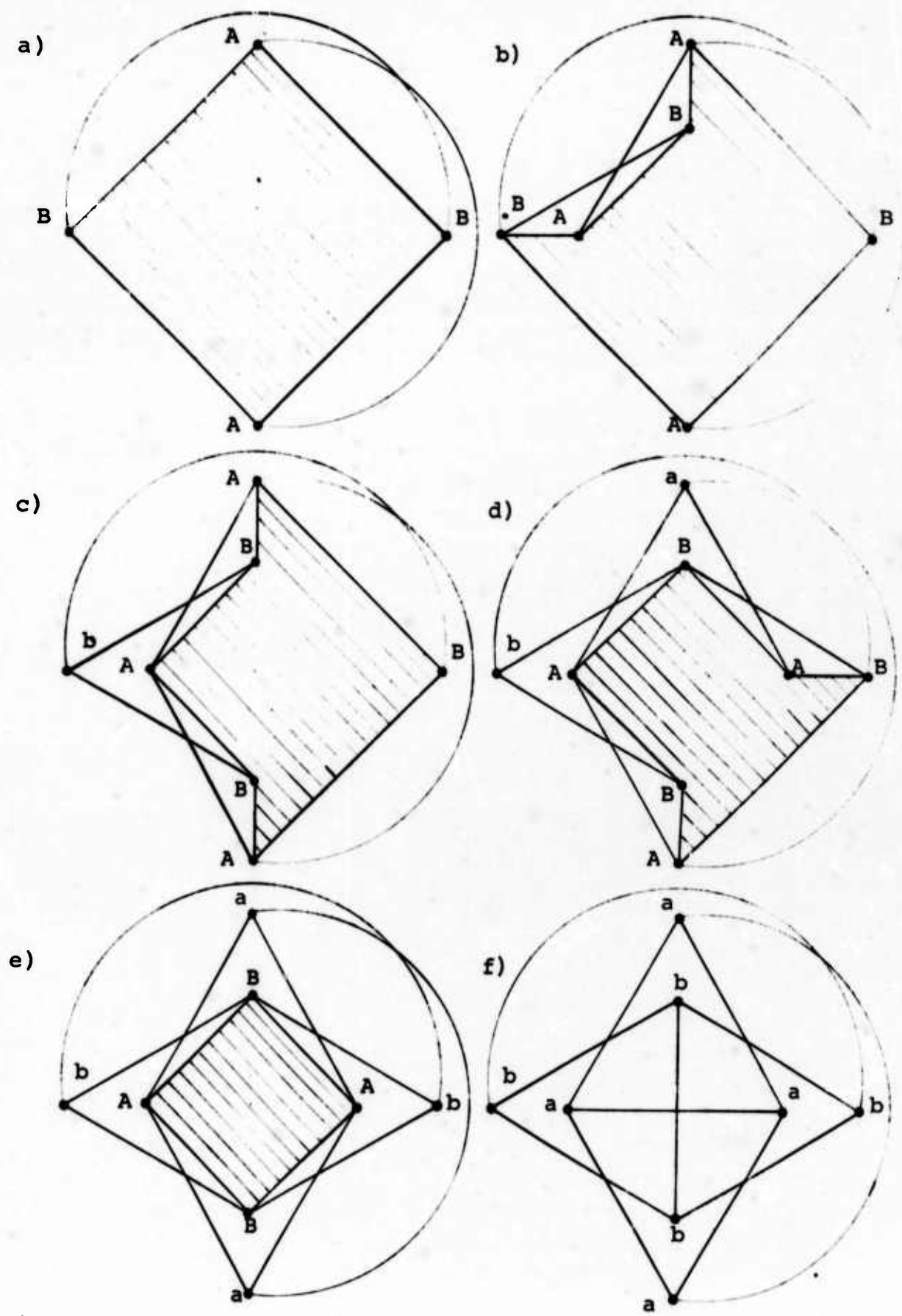
Fig. 7. An example of application of the web grammar
in Fig. 5

$$V_N = \{S, A, B\} \quad ; \quad V_T = \{a, b, c\} \quad ; \quad V_L = \{a\}$$

$$I = \{S, \; \underset{\bullet}{a} \; \underset{\bullet\!-\!\bullet}{a}\}$$

Rewriting rules:

(applies only if the upper
A-vertex and left B-vertex
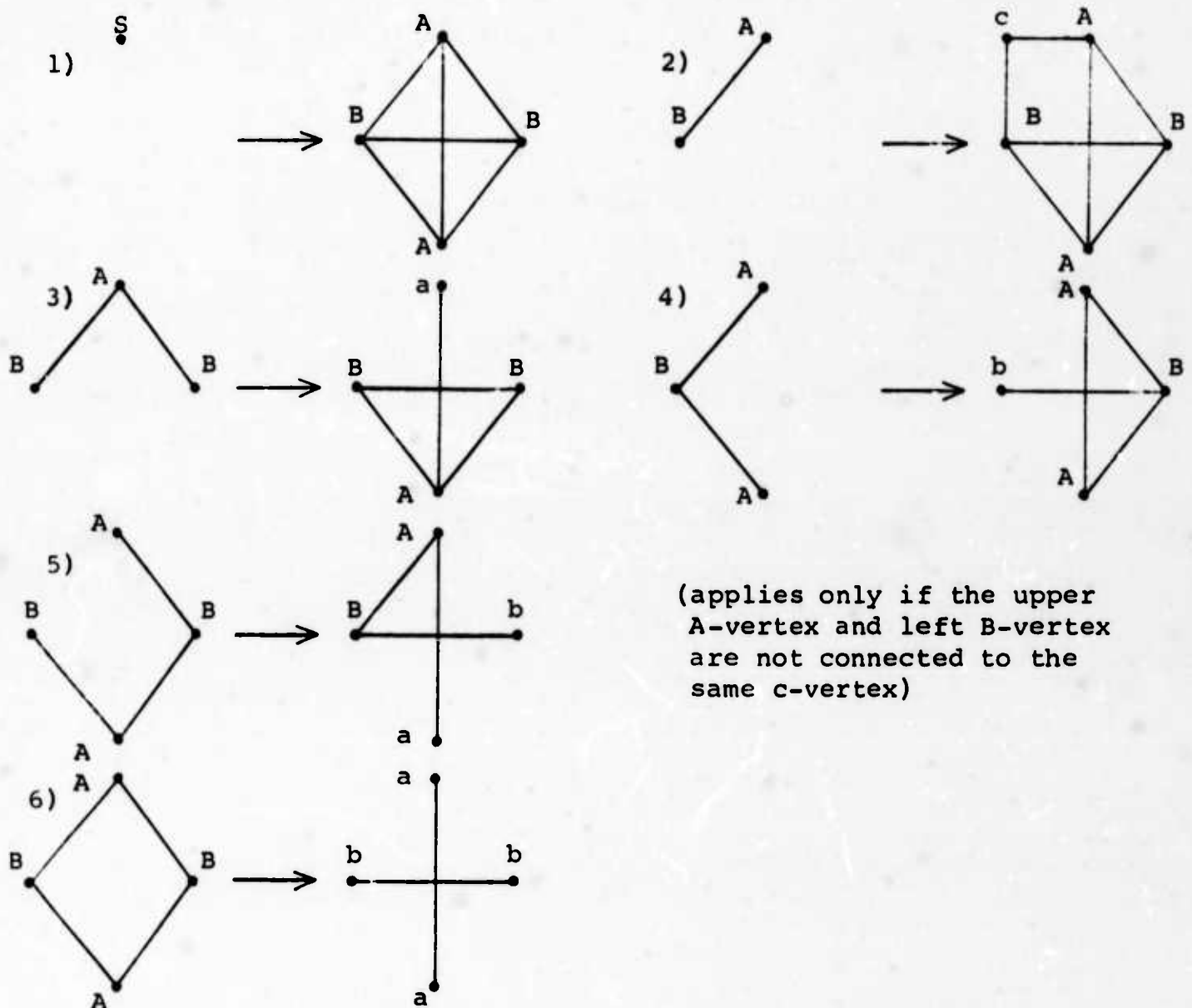are not connected to the
same c-vertex)

Fig. 8.   This normal, monotone web grammar indirectly
generates exactly all nonseparable planar graphs.
Note the applicability condition on rule 5.

$$V_N = \{S, A, B\} \quad ; \quad V_T = \{a, b\} \quad ; \quad V_L = \{a\}$$

$$I = \{S, \quad \overset{\bullet\!-\!-\!\bullet}{a \quad a}\}$$

Rewriting rules:
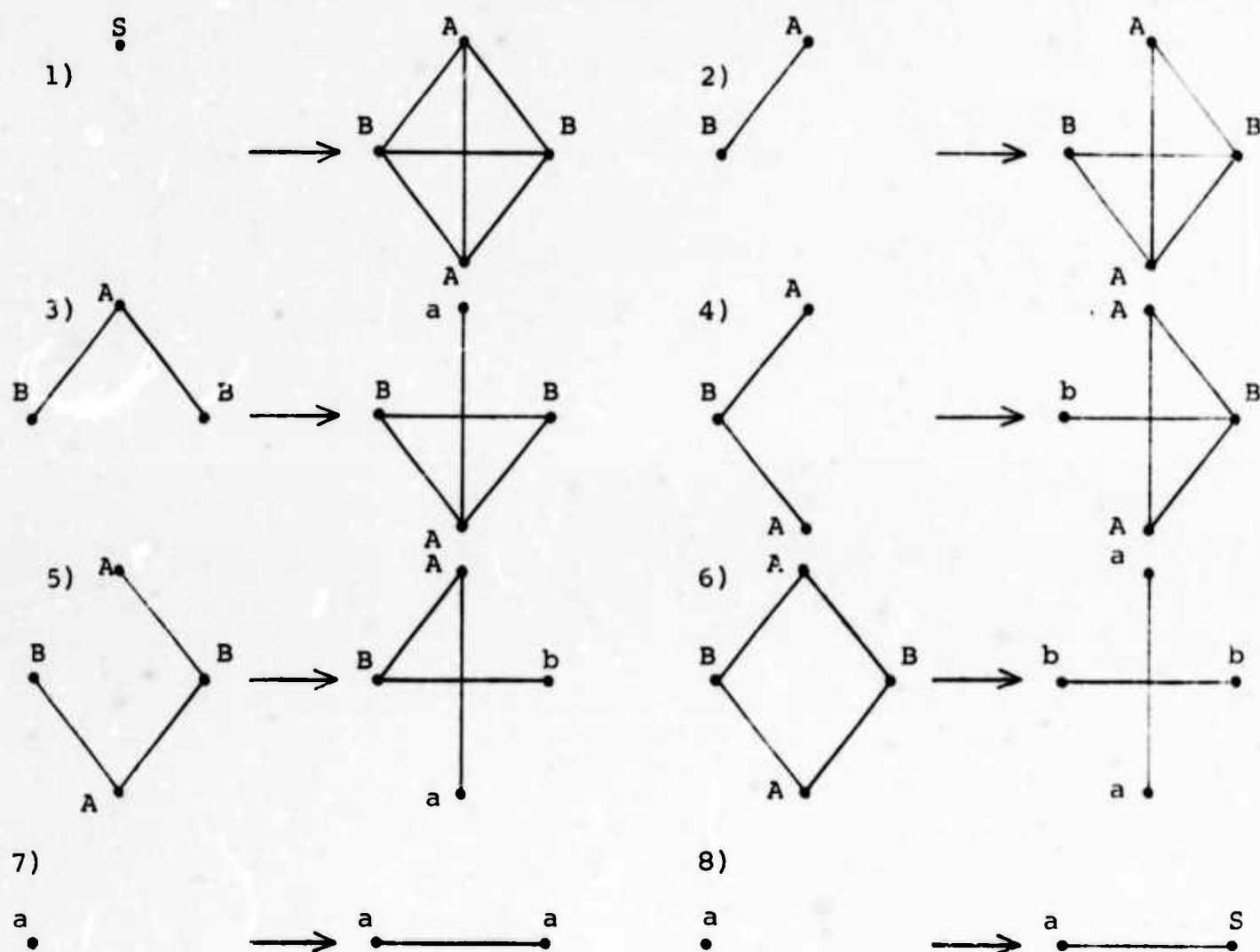

Fig. 9. This normal, monotone web grammar indirectly generates exactly all planar graphs

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1 ORIGINATING ACTIVITY (Corporate author)	2a REPORT SECURITY CLASSIFICATION
University of Maryland	Unclassified
	2b GROUP n.a.

3 REPORT TITLE

Separable graphs, planar graphs and web grammars

4 DESCRIPTIVE NOTES (Type of report and inclusive dates)

Technical Report

5 AUTHOR(S) (Last name, first name, initial)

Montanari, G. Ugo

6 REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
August 1969		6

8a CONTRACT OR GRANT NO.	9a ORIGINATOR'S REPORT NUMBER(S)
Nonr 5144(00)	
b PROJECT NO	Technical Report 69-96
c n.a.	9b OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
d	n.a.

10 AVAILABILITY/LIMITATION NOTICES

n.a.

11 SUPPLEMENTARY NOTES	12 SPONSORING MILITARY ACTIVITY
n.a.	Information Systems Branch Office of Naval Research Washington, D. C.

13 ABSTRACT

This paper is concerned with the class of "web grammars", introduced by Pfaltz and Rosenfeld, whose languages are sets of labelled graphs. A slightly modified definition of web grammars is given, in which the rewriting rules can have an applicability condition, and it is proved that in general, this extension does not increase the generative power of the grammar. This extension is useful, however, for otherwise it is not possible to incorporate negative contextual conditions into the rules, since the context of given vertex can be unbounded. A number of web grammars are presented which define interesting classes of graphs, including unseparable graphs, unseparable planar graphs and planar graphs. All the grammars in this paper use "normal embeddings" in which the connections between the web that is written and the host web are conserved, so that any rewriting rule affects the web only locally.

DD FORM 1473
1 JAN 64

14	KEY WORDS	LINK A		LINK B		LINK C	
		ROLE	WT	ROLE	WT	ROLE	WT

INSTRUCTIONS

1. **ORIGINATING ACTIVITY:** Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization (*corporate author*) issuing the report.

2a. **REPORT SECURITY CLASSIFICATION:** Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

2b. **GROUP:** Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. **REPORT TITLE:** Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parenthesis immediately following the title.

4. **DESCRIPTIVE NOTES:** If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. **AUTHOR(S):** Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. **REPORT DATE:** Enter the date of the report as day, month, year; or month, year. If more than one date appears on the report, use date of publication.

7a. **TOTAL NUMBER OF PAGES:** The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7b. **NUMBER OF REFERENCES:** Enter the total number of references cited in the report.

8a. **CONTRACT OR GRANT NUMBER:** If appropriate, enter the applicable number of the contract or grant under which the report was written.

8b, 8c, & 8d. **PROJECT NUMBER:** Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9a. **ORIGINATOR'S REPORT NUMBER(S):** Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9b. **OTHER REPORT NUMBER(S):** If the report has been assigned any other report numbers (*either by the originator or by the sponsor*), also enter this number(s).

10. **AVAILABILITY/LIMITATION NOTICES:** Enter any limitations on further dissemination of the report, other than those imposed by security classification, using standard statements such as:

   (1) "Qualified requesters may obtain copies of this report from DDC."

   (2) "Foreign announcement and dissemination of this report by DDC is not authorized."

   (3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through

   _____ ."

   (4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through

   _____ ."

   (5) "All distribution of this report is controlled. Qualified DDC users shall request through

   _____ ."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. **SUPPLEMENTARY NOTES:** Use for additional explanatory notes.

12. **SPONSORING MILITARY ACTIVITY:** Enter the name of the departmental project office or laboratory sponsoring (*paying for*) the research and development. Include address.

13. **ABSTRACT:** Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as *(TS)*, *(S)*, *(C)*, or *(U)*.

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. **KEY WORDS:** Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, roles, and weights is optional.